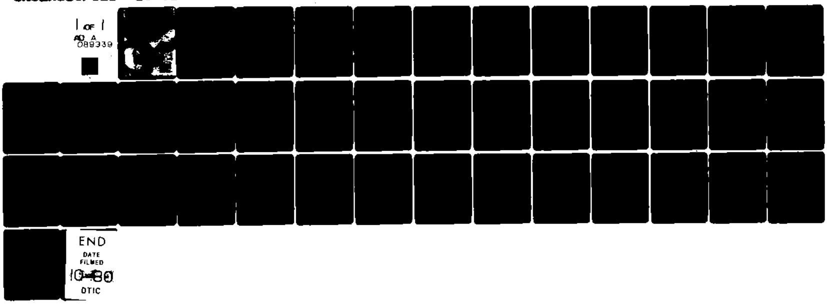
AD-A089 339 WHARTON SCHOOL PHILADELPHIA PA DEPT OF DECISION SCIENCES F/G 5/1
MODEL MANAGEMENT SYSTEMS: AN APPROACH TO DECISION SUPPORT IN CO--ETC(U)
AUG 80 J J ELAM, J C HENDERSON, L W MILLER N00014-75-C-0440
UNCLASSIFIED 80-08-04 NL

1 OF 1
AD A
089339



END
DATE
FILMED
10-80
DTIC

LEVEL

6

decision sciences

The Wharton School
University of Pennsylvania

MODEL MANAGEMENT SYSTEMS: AN APPROACH TO DECISION SUPPORT IN COMPLEX ORGANIZATIONS

Joyce J. Elam
John C. Henderson
Louis W. Miller

80-08-04

Department of Decision Sciences
The Wharton School
University of Pennsylvania
Philadelphia, PA 19104

This research was supported in part by ONR Contract No. N00014-75-C-0440.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 80-08-04	2. GOVT ACCESSION NO. AD A089389	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) MODEL MANAGEMENT SYSTEMS: AN APPROACH TO DECISION SUPPORT IN COMPLEX ORGANIZATIONS		5. TYPE OF REPORT & PERIOD COVERED Technical Report
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Joyce J. Elam Louis W. Miller John C. Henderson		8. CONTRACT OR GRANT NUMBER(s) N00014-75-C-0440
9. PERFORMING ORGANIZATION NAME AND ADDRESS Department of Decision Sciences The Wharton School University of Pennsylvania, Philadelphia, PA 19104		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research The Department of the Navy 800 N. Quincy St., Arlington, VA 22217		12. REPORT DATE August 1980
		13. NUMBER OF PAGES 34
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

model management, decision support systems, semantic networks

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This paper describes a system designed to extend the traditional DSS concept in order to support the model management requirements of complex organizations. Its objectives are to provide a mechanism to represent and to diffuse the organizational knowledge about models so that the user community can utilize this knowledge to adapt or build decision aids.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

MODEL MANAGEMENT SYSTEMS:

AN APPROACH TO DECISION SUPPORT IN COMPLEX ORGAMIZATIONS

1.0 INTRODUCTION

Recent years have seen a increased interest in developing interactive computer-based systems for supporting decisions that must be made in complex environments. Many of these systems are designed and built for decisions that relate to a specific problem [1, 10] -- portfolio management, manpower planning, etc. Each of these systems center around a single model that a decision maker can use to explore various problem characteristics and solutions. The model, the user interface, and the model solution process are tightly coupled into a self-contained system. As a result, such systems lack flexibility and are difficult to adapt when there are changes in the problems they are designed to deal with. The need for modification may be due to changes in the environment so that the problems to be solved are not quite like the ones in the past. Or, changes may be required because of learning on the part of decision makers -- new policies or goals or data may have to be introduced into the models and analytic framework of the DSS.

This paper discusses an extension of the decision support system concept that we term "Model Management Systems" (MMS). These systems support decisions relating to a variety of problems that arise in a complex decision making environment.

In particular, the major objectives of a MMS are:

(1) to facilitate the structuring of a decision so that analytical tools, possibly several in combination, can be used in generating possible solutions,

(2) to facilitate the use of the analytical tools that have been brought together through a structuring process.

Thus, rather than being a predefined decision aid, a MMS can be viewed as a system that dynamically constructs a decision aid in response to a particular problem. This is accomplished by drawing on a knowledge base of models that captures the technical expertise of a management scientist plus an understanding of the basic activities involved in a given decision making environment and the way in which these activities interrelate.

This knowledge can be diffused throughout the decision making environment and adapted as necessary to support a decision maker in structuring as well as analyzing a problem. Knowledge representation, diffusion of knowledge, and adaptation of this knowledge in solving problems are basic characteristics of a MMS.

A MMS draws upon and synthesizes concepts from management information systems, computer science, artificial intelligence, quantitative methods, and organizational behavior. References to some of this material can be found in [2, 6, 14].

The remainder of this paper will discuss the MMS concept in depth. Organizational factors that have created a need for a MMS are discussed in the next section. The decision making environment which we see as being appropriate for a MMS is discussed in Section 3. Section 4 discusses the user roles involved with the MMS, and Sections 5, 6, and 7 build upon our experiences with prototype systems to discuss a structure of each MMS component. Section 8 presents our conclusions.

2.0 MODEL MANAGEMENT: WHY IS IT NEEDED?

A common characteristic of all decision makers is the use of a "model" as a basis to gather data, analyze this data, and eventually make a choice. These models may be intuitive or externalized, i.e., formulated in some symbolic manner. Even those models that have been externalized may not be in a form that allows the utilization of computer technology to aid in processing. Nevertheless, a primary task of all decision makers involves the building and/or storing and/or recalling and/or executing of models.

Further, as decision environments become more complex, the need for formally externalized models increases. Such models greatly extend the information processing capabilities of the decision maker, providing a mechanism to significantly increase his or her effectiveness. To a large extent, this has been the reason for the growing interest in Decision Support Systems (DSS). Keen and Scott Morton [10] define DSS as a system, normally computer based, which supports decision makers who are dealing with semi structured problems [13]. The focus on semi structured problems is crucial. In effect, it requires the role of the DSS to be one of enhancing a process rather than providing a "product" or an answer. The goal is to create a more effective decision maker by facilitating his or her ability to search, create alternative solutions, and evaluate these solutions through the use of models. Such a system must have the capability to model the specific decision or problem under consideration and even to fine tune the model to match the style of the manager. In most decision support systems, this is achieved by building a problem specific tool that contains a single model. As a result, there is little capability to use the decision support system for other applications, even if these applications are closely related.

Most of the recent technological developments in DSS have been directed toward providing software and hardware that reduce the cost and time necessary to build and implement problem specific decision aids. While the benefits from problem specific aids have been demonstrated [1, 10], we contend that, in many situations, significant benefits can be realized by developing more generalized decision aiding systems that access, utilize, adapt, integrate models.

For example, an organization may be pursuing the goal of developing a corporate financial planning model. A few years ago, this model would have most likely been developed and operated by a centralized planning group for the managers in the corporation. We would not consider such a model as being part of a decision support system since, without the active involvement of the managers in developing and using the model, the impact of the corporate model on their planning processes would be minimal.

Today, however, commercially available interactive financial planning systems allow non-technical managers throughout the corporation to build and analyze financial models in their areas of responsibility. Moreover, these systems enable a manager to create his or her own unique decision aid for analyzing the problem represented by such a model. The decentralization of modeling activities made possible by these systems has several benefits: expertise that exists in various functional areas can be utilized directly in the building, analysis, and interpretation of models; managers gain insights into their particular planning problem and thus, it can be argued, make better decisions.

While decentralization has its advantages, it creates difficulties when one wishes to view the overall corporate model in terms of its individual submodels, and particularly, when one wishes to use the knowledge represented in the corporate model to analyze problems that cut across organizational boundaries. To overcome these difficulties, management and coordination of the collection of models that comprise the corporate model become essential.

Without management and coordination, the application of a model often appears unique and is developed in response to a special information need or to support a unique decision. Development of a model is usually justified by the model's impact on an immediate decision. Justification seldom includes utilization of the model in a variety of contexsts. Hence, there is little incentive to integrate the model into an overall system. Moreover, the decentralization of model building, ownership, and use creates a potential for inconsistent definitions and representations. This leads to two types of errors in matching models to decision problems: an appropriate model exists but is not recognized, or an existing model is inapproriately applied.

The development of a systematic model management system for a corporate financial planning model, or for any other problem environment involving numerous models, is not a simple task; but the costs of <u>not managing</u> the model resource are real and are becoming more significant. Perhaps the most obvious cost is "reinventing wheels." A more significant cost is that the knowledge gained in developing a model is often not available to others in the organization. The result can be a decision <u>not to use a model</u> because:

(1) there is no memory of organizational modeling activities and/or

(2) no mechanism to diffuse and adapt such knowledge in the organization exists.

Even when one has knowledge of separate models that can be integrated to form a particular decision aiding system, the integration of models often requires tedious, time consuming manual interfacing. As a result, the decision is often <u>not to use a model</u> at all or to <u>develop a new model</u>.

In addition, there is often little or no concrete basis for estimating development costs, assigning priorities to subtasks, or selecting personnel (perhaps from another part of the organization) for a modeling activity. The inability to resolve such issues due to a lack of "memory" may result in an inflated perception of costs and time leading to a decision not to develop a model.

All this argues strongly for viewing models as a corporate resource that needs to be effectively managed. The MMS provides this management.

3.0 MODEL MANAGEMENT SYSTEMS: THE ENVIRONMENT DIMENSION

The Keen-Scott Morton definition of Decision Support Systems has its origins in the Management Information System framework suggested by Gorry and Scott Morton [8]. They developed a two dimensional framework shown in Figure 1 involving the characteristics of the decision process and nature of managerial activities.

As indicated earlier, and noted by Keen and Scott Morton, a DSS must be mapped closely to the decision process itself. As a result, most of the existing DSS applications are unique management tools that offer little opportunity for transfer to other application areas. Transferability has been recognized [4] as an important issue in the design of support systems, particularly, when one considers the impact of environmental complexity.

In a complex environment, Thompson and Dill [5, 15] suggest that organizations have the following characteristics: many boundary spanning units (and subdivision within these units), a technical core that is layered or differentiated, and responses to environmental influences that involve individual localized monitoring and adaptation. Hence, information sources and flows are ill-defined with many interfaces required between organizational units and individuals.

In the context of the application of DSS, this has significant impact. The complexity of the environment can be viewed as a third dimension on the Gorry-Scott Morton framework. In a simple environment, the organizational subunits involved will be relatively few with well defined interfaces and coordinating mechanisms will be relatively simple. Each DSS can be effectively viewed as a unique application of a single model. Since there are few organizational subunits, the knowledge gained during an application can be diffused easily.

In contrast, a complex environment has ill-defined information flows and many organizational interfaces, i.e., increased boundary spanning units and differentiation in the technical core. Further, the dynamic nature of such environments results in constantly changing models and associated information flows. Each subunit models unique aspects of its environment and continually adjusts these models. Hence, the subunit interfaces are poorly defined.

In addition, knowledge required to understand the decision process tends to be localized to individual subunits. The problems of diffusion and integration of this knowledge base are substantial. To appropriately deal with the complexities of interpersonal and model to model communication, the organization must have a systematic means to capture the model related knowledge, to access this knowledge base, and to use the knowledge to support decision making. The MMS is intended to meet this need.

4.0 USER'S ROLES IN MMS

A MMS supports the user community in its attempts to interact with the computer for problem solving. But, who is the user community? Rather than focus on individuals or their specific organizational titles or functional responsibilities, we will focus on organizational roles often identified with the design and implementation of a DSS. Alter [1] identified 5 key roles associated with a DSS, the user, decision maker, intermediary, maintainer, and feeder. Adapting his classification, we identify 4 key roles associated with the MMS: decision maker, model user, model maker, and model implementor.

The decision maker is the client for whom the particular DSS application is being designed. It is his or her effectiveness that the DSS is ultimately supposed to serve. As such, the world view of the individual(s) in this role will have a primary impact on problem definition and the eventual success of the DSS.

The model user interacts directly with a dynamically constructed decision aid, running cases and interpreting outputs for the decision maker. The model user also requests the model maker to revise the decision aid as necessary.

The model maker uses knowledge of problem solving technology, the application of this technology to problems within the organization, and the particular decision context to conceptualize an appropriate logical structure for the problem. Based on this conceptualization, the model maker oversees the creation of a decision aid which is an integration of selected models from the knowledge base. If this conceptualization requires a model that does not currently exist in the knowledge base, the model maker calls upon the model implementor to create an appropriate model, which is then added to the knowledge base.

The model implementator is responsible for translating the specifications supplied by a model maker into appropriate computer programs and for integrating these programs into the MMS.

In some environments, one can expect a single individual to fill more than one role. In other environments, a single role will be filled by several individuals. In any event, a person in one role will invariably be concerned with some aspects of other roles. The MMS provides mechanisms to support each role as well as interfaces between roles.

The major activities supported by the MMS are shown in Figure 2. Also depicted in Figure 2 are the outputs from the activities and the roles associated with the activities. Figure 3 illustrates the five basic components of a MMS: knowledge base, interrogator, builder, analyzer, and executor. The remainder of this paper discusses how each of these components are structured in order to support the activities outlined in Figure 2.

5.0 KNOWLEDGE BASE

A major objective of a MMS is to facilitate the structuring of a decision so that analytical tools can be used in generating possible solutions. In order to perform this objective, the MMS must have "intelligence" that it acquires by accessing knowledge - both general problem solving knowledge as well as more specialized knowledge about the problem environment. In addition, when the MMS acquires new knowledge about the problem environment from its interaction with users, the MMS stores this knowledge so that it can be accessed at a later time. Thus, a key component of the MMS is a knowledge base.

In this section, we discuss two major issues involved with the knowledge base. The first issue is what types of information should be represented in the knowledge base, and the second issue is how this knowledge should be represented.

The MMS must represent the technical problem solving knowledge that one would expect a management scientist to have (traditional view of the analyst). This knowledge involves information on mathematical model types, on the parameters that define one model type and distinguishes it from other model types, and the structural relationships between parameters. The MMS must also have an understanding of the basic activities involved in the problem environment and the way in which these activities interrelate (traditional view of the user). In order to be able to converse with its users, the MMS must also have knowledge of an appropriate vocabulary that is understandable to the users. Finally, the MMS must have knowledge about models that have been developed for specific applications (which will be referred to as a model instance) in order to support the execution and analysis of these models. The knowledge base, thus, contains four types of information: technical, application, language, and model instance.

More importantly, the knowledge base contains information on how all of this information is related e.g., how is a user-supplied description of a problem related to basic activities that are known to exist in the problem environment and how can the activities be structured into an appropriate model to which some algorithm can be applied. It is this information that allows the MMS to bridge the gap between user and analyst.

The second issue to be discussed concerns how this "knowledge" should be represented. In choosing a representation, it must be recognized that this knowledge is highly interconnected and forms a network of concepts, facts, and perceptions. Also most of this knowledge is abstract in nature; the only concrete knowledge being the collection of model instances.

Traditional data models were rejected as the vehicle for representing the knowledge base for two reasons. First, we are in a relation rich, data poor environment. That is, we wish to represent diverse and detailed abstract knowledge about relatively few entities. Data models are designed to represent mostly concrete knowledge about comparatively smaller and more uniform sets of details for a large number of entities.

Secondly, the processing of abstract relationships between entities will be the primary way in which the MMS acquires "intelligence." The representation should have an explicitly defined set of these relationships ("is a part of", "is analogous to", "is the same as except","is a kind of", etc.). Conventional database models allow one to define only concrete types of relationships. Any meaning attached to a relationship must be imbedded in the user's processing program.

These factors prompted us to look to the field of artificial intelligence for representation models more suited to the needs of a MMS [7, 12]. The Structured Inheritance Network (SI-Net) developed by Brachman[2] was chosen as the representational model for the MMS knowledge base.

A SI-Net is a graphical language composed of nodes and links for describing concepts and the interrelationships between these concepts. A "concept" is defined as a set of functional roles tied together with an explicit structuring relationship. Two basic types of relationships are involved in defining a concept "is a part of", which is represented by a DATTR link and describes the functional roles in a concept, and "are structured as", which is represented by a STRUCTURE link and describes how these roles are put together. Each role is described by (1) a role name, (2) a value restriction that represents the types of things that can fill this role (referred to as a role filler), (3) a number which represents how many role fillers there are, and (3) a modality value that represents whether the role must have a filler supplied externally (necessary), or a role filler is not required (optional), or the role filler is to be derived from a processing routine or from the structural relationship defined for the concept (derived). Figure 3 illustrates the basic SI-Net notation for the concept "shipping point.".

A SI-Net also represents relationships between concepts. These relationships, which are represented by named links, include such things as "is analogous to", "is a subconcept of", "is the same as except", "is an individual of", etc. These links allow a concept to inherit all properties (roles and structure) of other concepts, and to modify, extend, or differentiate these properties as necessary.

The MMS knowledge base can be thought of as four distinct, but coupled SI-Nets: the technical net, the application net, the language net, and the model instance net. (We represent model instances in the SI-Net for consistency, although the model instances can be effectively represented in a data base model.) Figure 5 illustrates a small portion of a technical net and an application net. In the technical net, the concept "network" represents technical information about a type of mathematical programming model. It is defined in terms of the roles "nodes", "arcs", and an "objective function." Note that the value assigned to objection function is derived; this represents the fact that the role filler for objective function must be derived from some processing routine (e.g., a network optimization routine). The concept "network" also has a structural condition which specifies among other things that the network model must be connected. The concept "arc" is further defined by roles such as "from node", "to node", "cost", "flow", etc. Like the technical net, the application net also consists of concepts and structural conditions. These concepts are linked together in a hierarchy of less-to-more abstract concepts. The less abstract concepts relate to fundamental activities, while the more abstract concepts focus on problem systems. In Figure 5, the activities of storing and shipping are represented. Structural conditions are used to cluster related activities into a problem system. For example, a distribution system has shipping and storing, where the shipment of goods from locations must also involve the storing of the goods at some of the same locations.

The language net directly confronts the particular jargon, terminology, etc. associated with an organizational environment. The language net provides the capability for translating between a user's description of a concept and the system-defined label associated with the same concept. Thus, a manufacturer speaks of factories, products, and warehouse. While the military communicates in terms of depots, spares and loses. They both may be referring to identical problem systems and technical structures. The language net allows us to communicate to the user in a language that is both familiar and more precise.

The reason for choosing the SI-Net representation was that this representation supports the types of processing that will allow the MMS to act intelligently. The characteristics of a SI-Net that are particularly important to such processing are discussed below.

The SI-Net permits knowledge to be "chunked" into groups of descriptions of closely associated entities rather than requiring knowledge to be lists of independent facts. Being able to associate concepts will allow the MMS to perceive, for example, that "shipping" is a kind of "process" that, in some situations, may be equivalent to an "arc" in a "network" model.

The SI-Net language defines a set of abstract relationships that are sufficient for processing routines contained in other MMS components. For example, two mathematical models defined in the technical net can have their structural differences represented by a difference relationship. This relationship provides the basis for the MMS to ask questions of the user about a particular problem. The user's responses enable the MMS to select an appropriate model structure.

The SI-Net representation allows new concepts to be derived from old concepts. Thus, the MMS should have the capability to perceive new structure in terms of already known concepts, i.e., to learn. In particular, the MMS should be able to make inferences. For example, if a set of problem activities can be represented as a "network" model and a "network" model inherits all properties of a "linear programming" model, the set of problem activities can also be represented as a "linear programming" model.

6.0 THE INTERROGATOR

The knowledge base is made available to the organization via the interrogator component. This component functions somewhat like a data retrieval system but has enhanced capabilities due to the SI-Net. The basic objectives for this component of the MMS are fourfold:

(1) to recall actual model instances

(2) to support the problem definition and model conceptualization process

(3) to identify necessary modifications for existing models

(4) to initiate the actual process of model building

The attainment of these objectives is realized through an interactive stimuli/response sequence with a user. The system attempts to identify basic concepts, verify these concepts and infer potential logical structures that could be used to model the problem. Further, user-generated labels for concepts are captured and used to dynamically adapt the semantics of the interaction so as to become increasingly "friendly."

Normally, non technical users prefer to initiate interaction with the system through the application net. The interaction begins with a user responding to a general question such as 'What are the important activities/actions associated with your problem?'. The response is free format, with the system processing the response to identify candidate concepts, generating questions to verify the interpretation of concepts, and using the SI-Net to infer possible problem systems. For example, if the concepts of shipping and storing have been identified as appropriate and relevant, the system can infer via structural conditions that a potential problem system of distribution may exist. If the structural conditions for such aggregate concepts pose special or unprocessed conceptual relationships, these are investigated. Similarly, if the user initialy responds at a relatively abstract level, i.e., that the problem activity is distribution, the system can deduce required concepts and investigate the extent to which they are defined, appropriate, and relevant.

One advantage of initiating the interaction via the application net is the ability to use labels for basic action concepts and their roles. For example, during the verifying process for the concept "ship", the user may indicate that the "places" (a basic concept) from which goods are shipped have labels of "warehouse", "plant", etc. This allows the system to adapt the interface dialogue to utilize terminology which is more concrete in the user's mind.

The labels generated are linked through the SI-Net structure to more general concepts. Thus, "warehouse" is associated with "shipping location". This allows the system eventually to have the user create an instance of his or her problem by entering examples of relationships between identified concepts. For example, the user is asked to generate shipping routes (i.e., matched to and from labels). This "prototyped" model is examined and the technical net is used to infer possible logical structures.

The problem structuring process is iterative. It may require returning to the application net to verify new concepts or to clarify inconsistencies in the use of concepts. As interaction process proceeds, the results are an evolving model definition.

The Interrogator, which operates with the technical, application, language, and model instance nets, contains primitives that are the basis of commands allowing users to recall, store, compare, and contrast concepts.

The compare primitive operates in two mode: hypothesizing and discovery. In the hypothesizing mode, the "comparison" is between the MMS's understanding of concepts (i.e., model structures, activities, etc.) and the user's perceptions of the same concepts. In the discovery mode, the "comparison" is between two concepts that are already represented in the knowledge base. Hypothesizing suppports problem structuring. The problem structuring process is initiated by the MMS with questions such as "what are the actions associated with your problem?" Discovery permits a user to determine if the system's understanding of a concept is the same as his own. The discovery mode is initiated by user commands such as 'recall an activity like production' or 'compare model 1 with model 2.' Finally, constrast primitives support the use of commands such as 'find a counter example' or 'negate assumptions' or 'contrast model 1 with model 2.'

As indicated in the discussion of objectives, the initial interrogation process forms a foundation for the building component. The requirements to execute the building and analysis components are discussed in the next section.

7.0 BUILDING, EXECUTION, AND ANALYSIS

Interrogation of the knowledge base results in the model maker having identified a useful model or set of models that will comprise the decision aid. Identification of useful models in itself is not sufficient to get a running program. The transition between model identification and a program is done in the building process under the supervision of model maker. The result of building is a decision template that contains human and machine readable specifications to instruct the executor (operating system) component of the MMS in preparing a computer program to implement the decision aid.

The execution phase, based on the specifications in the decision template, includes physical accessing of computer code from libraries, linking together models (in the usual sense of preparing a load module), executing the program, and possibly saving the outputs for further processing. Somewhere between building and execution, a top level control procedure for the decision aid must be generated. We prefer to make this part of execution rather than building in order to keep decision templates simple.

Analysis means presentation of results to the user and possibly includes processing outputs of models by statistical and graphical procedures. For some applications, software support for analysis is an integral part of models. But frequently, it is advantageous to perform analysis separately from executing models. Therefore, the MMS contains an analysis subsystem.

By now there are many software systems that support building, execution, and analysis and supply some services similar to those of the MMS. The several commercially available financial planning systems comprise a class of typical systems. Common characteristics of such systems, which we shall call "closed" systems, are that they are focused on specific problem areas, and models are constructed in special purpose languages that are supplied with the system. Models being managed by the MMS, however, are programmed in standard languages that are not themselves portion of the MMS. This means that the part of the MMS that supports building and execution takes on aspects of a more general computer operating system. The MMS is not meant to be restricted to a specific problem domain or particular kind of model. Closed systems are intended to allow all or most roles, as described in Section 4, to be taken on by the same individual. In particular, the model maker is also the implementor. With the MMS, however, we expect many models to be large and complex, so that the services of professional implementors are required and the full power of standard programming languages can be employed. We expect that model makers will do very little that resembles computer programming.

Building. The purpose of building is to organize enough information so that the models chosen in the interrogation process can be run. In order to do this, three kinds of issues have to be dealt with. They are: (1) inter-model communication, (2) sequencing and control of models' executions, and (3) obtaining data from the user.

To help resolve these issues, the MMS has access to a collection of information beyond that already discussed as components of the knowledge base. We call this information "model documentation," and it exists specifically to support building. Documentation for a model is written in a special "documentation language" by the implementor as part of the model coding process. Documentation language statements are interpreted and used by the MMS, and their content can be displayed in order to support the model maker role. Documentation language statements mainly describe the inputs and outputs of a model, but the deal with other topics as well (such as providing names for accessing program libraries).

Inter model communication is a problem because the outputs of some models are inputs to others. Thus, part of the building activity is to identify the sources of each model's inputs as coming from another model or from an outside data base. It is also necessary to arrange for compatibility in that data going from one model to another must be or put in a form that can be understood by the recipient. Reorganizing and reformatting the output of one model to be consistent with the input requirements of another model is done by incorporating programs that appear to the MMS's executor as though they are simply additional "models" to be included in the decision aid.

Organizing data flow between models is supported by the documentation language statements for the models. If model AGGREGATE PLANNING requires as an input a vector called MONTHLY FORECAST, the MMS will try to find another model (among those that are to be combined) that has a similar output. If a match is found, the model maker has an opportunity to verify that the linking of these particular items is appropriate. In this case, the model maker has a good idea of what the correct linkage should be because of information gained from the interrogation process. The model maker also has access to the documentation language statements, which can include descriptive material for the model maker's benefit.

Sequencing and control of models means deciding in what order the models should be called upon and arranging for iterations of model executions, should that be required. Requirements for sequencing models are derived from considerations of inter model communication. (In the example of the forecasting and planning models, it is clear that the forecasting model has to be run first because the planning model uses its output.) Iteration means repeating the execution of models or groups of models in order to achieve some kind of convergence or to generate results for sensitivity analysis. Specifying requirements for iteration is the closest that the model maker comes to performing an activity that resembles computer programming.

User's Data. In addition to models communicating with each other, models frequently require information from users. Inputs from users can include parameters representing problem data, choices among options available within models, or names of files that will serve as sources of input data. Managing users' input data is an important part of the model management concept in that programs run under the MMS should not interact directly with the user through normal read statements. Instead, the model implementor describes in documentation language statements what is wanted from the user. The description includes text for prompts, help texts, and error conditions that should be checked for. Then the model management system will prompt the user and make the information available to the model through "get" procedures called from the model program.

There are several reasons for taking this indirect approach to handling users. First is that programming high quality user interactions is difficult, and the MMS can provide better and more uniform interactions than most programmers would be willing to implement. Another benefit is that models may require a great deal of data from the user, but most of the data won't be changed over a series of model executions. The values of such data items can be given by the model maker to become part of the decision template. The remaining data inputs are defered until the model is executed. The third benefit is that by having users' data pass through the MMS, the data can be saved (and even annotated) as a form of documentation.

Checking. Checking the consistency of the linkages in a model implied by a decision template is a valuable service provided by the MMS. The checking procedure verifies that the models are being executed in a feasible sequence and that every model's inputs can be found. Reports produced by the checking process can be saved as additional documentation of the decision aid, and can be examined for further verification that the decision aid makes sense in terms of the problem to be solved. Checking is a dry run for executing the model, and it employs procedures that are part of the MMS's facility for generating the decision aid's controlling routine.

Saving and Modifiying Decision Templates. Once the effort has been expended to produce a decision template, the template can be saved in the knowledge base. Facilities are provided to make modifications to old decision templates so that models specified by them can be removed or added and user's data can be changed.

Analysis of Model Outputs. As mentioned earlier, in many applications it is advantageous to carry out analysis in a process separate from running models. As an example, we have applied many model management concepts to micro-analytic simulations [11] that simulate the interaction of events and policies on samples of individuals. The models take a disaggregated view and produce voluminous outputs, giving a record of attribute values for each of the simulated individuals. By having aggregation and analysis done on the results of models rather than by the models themselves, users can explore results in many different ways without having to rerun models. Furthermore, having a separate analysis subsystem makes it possible to employ analytic prcedures to make comparisons across the results of several model runs. The analysis system used in this work resembles an interactive, extensible statistical package, and its design is consistent with many of the general model management concepts, including a library of modules that can be added to, a template to describe what is to be done to what data, and user interactions as described previously.

Many of the ideas in this section have been operationalized in WHIMS [9], which was developed in order to provide flexibility and modularity in working with micro-analytic simulation models. Although WHIMS is weak in supporting the interrogation process and it is quite restricted in the kinds of model structures that it supports, the building, execution, and analysis subsystems are highly developed.

8.0 CONCLUSIONS

This paper describes a system designed to extend the traditional DSS concept in order to support the model management requirements of complex organizations. Its objectives are to provide a mechanism to represent and to diffuse the organizational knowledge about models so that the user community can utilize this knowledge to adapt or build decision aids.

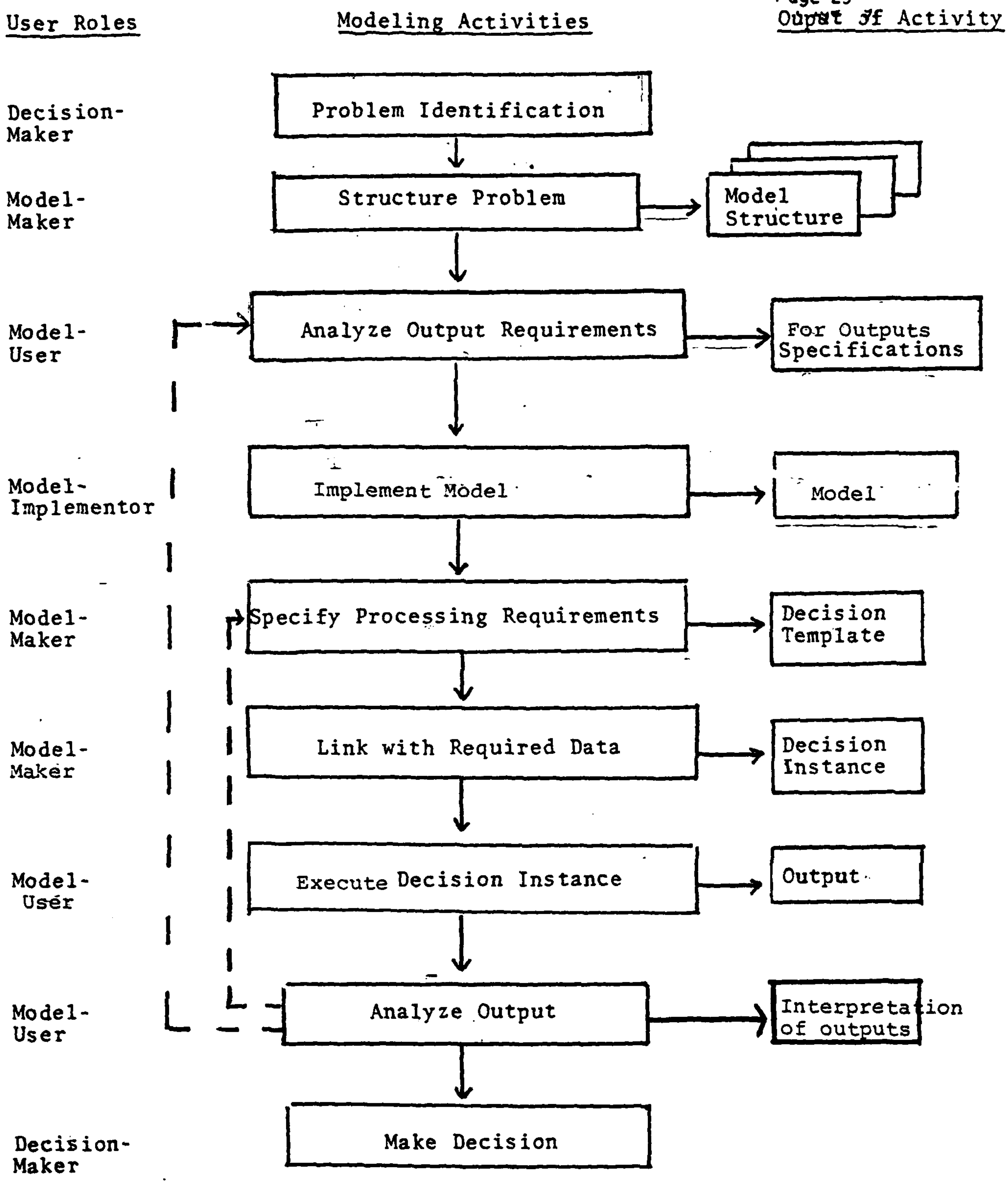
The model management system consists of five components: knowledge base, interrogator, builder, analyzer, and processor. The SI-NET formalism is used as the data model for representing models, actions, language and their relationships. The interrogator uses the structural links of the SI-NET formalism to implement procedures that recall, compare, or contrast concepts. The building, analysis, and processor components take the initial model definition provided via the interrogation process and create and run an actual decision aid. Building the decision aid requires not only identifying existing model resources but also linking them to the appropriate data bases, controlling the sequence of their execution, and continually checking for consistency in the linkage process. A documentation language is provided to support the building process so that the knowledge of modifications and extensions to existing software modules or the creation of new software modules can be quickly provided to the user community and, eventually, incorporated in the knowledge base.

The goal of the MMS is to provide a flexible, diverse Decision Support System. It recognizes the existence of multiple roles associated with model building and implementation and attempts to coordinate the activites of each role so that they can effectively operate in a decentralized setting. In essence, it is a means to accumulate, utilize, and manage the expanding organizational knowledge relating to the building and use of computer-supported decision aids.

Type of Decisional Task	Management Activity: Operational Control	Management Control	Strategic Planning	Support Needed
Structured	1 Inventory reordering	4 Linear programming for manufacturing	7 Plant location	Clerical, EDP or MS models
Semistructured	2 Bond trading	5 Setting market budgets for consumer products	8 Capital acquisition analysis	DSS
Unstructured	3 Selecting a cover for Time magazine	6 Hiring managers	9 R&D portfolio development	Human intuition

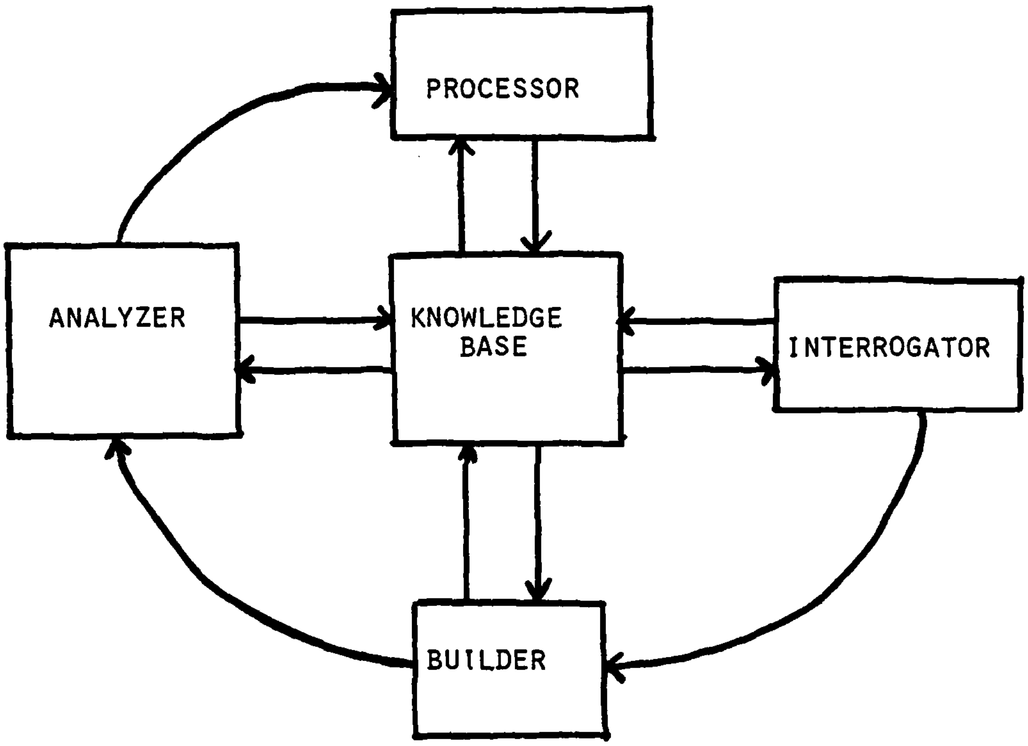
Framework for Information System

FIGURE 1

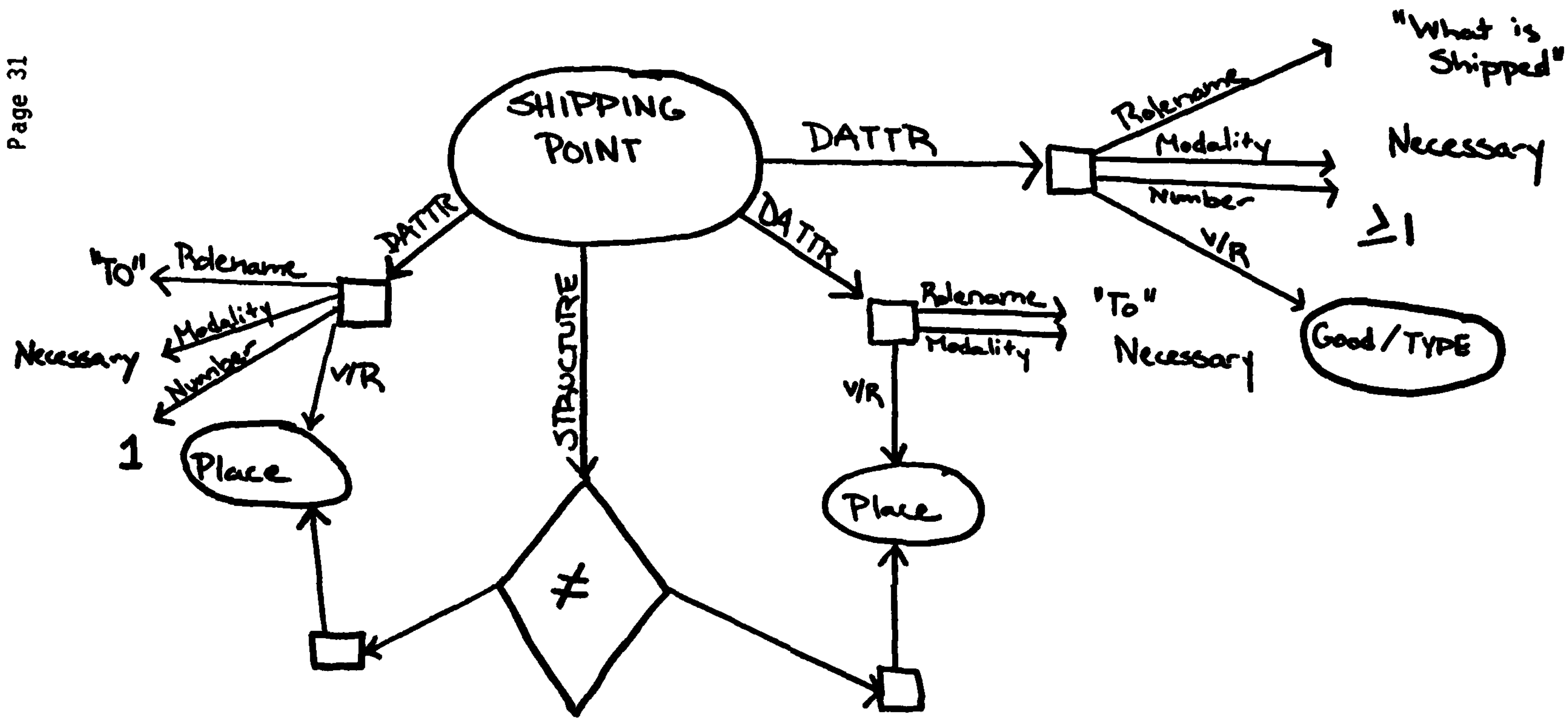


MODELING ACTIVITIES

FIGURE 2



MODEL MANAGEMENT SYSTEM
FIGURE 3



Simplified SI-NET Representation of "Shipping Point"

FIGURE 4

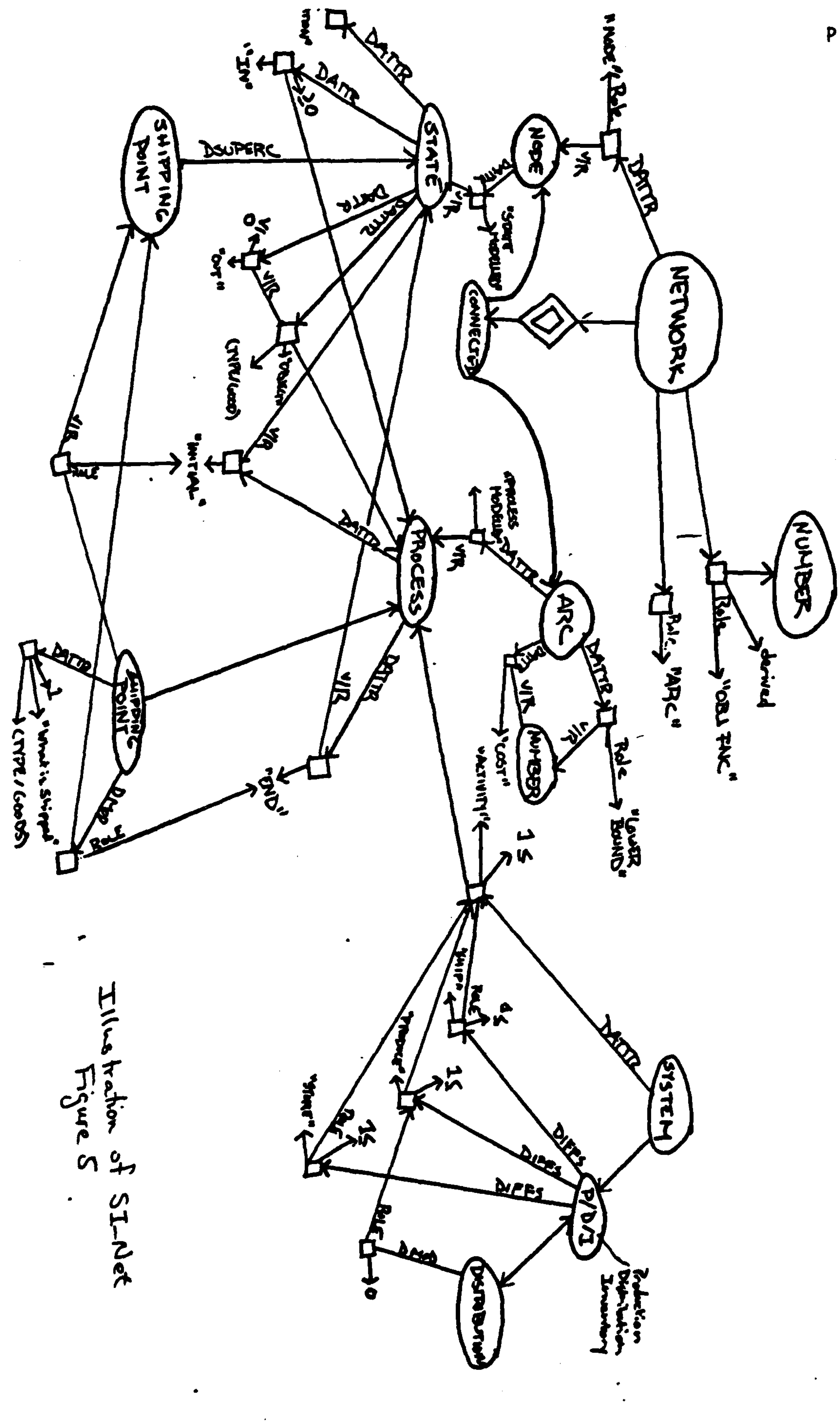


Illustration of SI-Net
Figure 5

Bibliography

1. Alter, Steven L. Decision Support Systems: Current Practice and Continuing Challenges, Addison Wesley, 1980.

2. Bonczek, R. C., Holsapple, W., and Whinston, A., "Computer Based Support of Organizational Decision Making", Decision Sciences, April, 1979.

3. Brachman, R.J., "A Structural Paradigm for Representing Knowledge," Report No. 3605, Bolt, Beranek, and Newman, Inc. 1978.

4. Donovan, John, "Database System Approach to Management Decision Support", TODS, 1,4, pp. 344 369, 1979.

5. Dill, L. and Hilton, W. Reitman, The New Managers, Prentice Hall, N.J., 1962.

6. Elam, J. "Model Management Systems: A Framework for Development", Working Paper 79 02 04, Department of Decision Sciences, University of Pennsylvania, 1979.

7. Finuller (Ed). Associative Networks: Representation and Use of Knowledge by Computers, Academic Press, 1979.

8. Gorry, G.A., and Scott Morton, M.S., "A Framework for Management Information Systems," Sloan Management Review, vol. 13, no. 1, pp. 55 70, Fall 1971.

9. Katz, N. and Miller, L., "An Interactive Modeling System," Working Paper 77 09 02, Department of Decision Sciences, University of Pennsylvania, 1977.

10. Keen, P. and Scott Morton M., Decision Support Systems:, An Organizational Perspective, Addison Wesley 1978.

11. Orcutt, G.H., Greenberger, M.H., Korbel, J., and Riolin, A.M., Microanalysis of Socialeconomic Systems: A Simulation Study, Harper and Brothers, 1961.

12. Quillian, Ross, "Semantic Memory", Semantic Information Processing, Marvin Minshy, ed. MIT Press, 1968.

13. Simon, H.A., The Shape of Automation for Men and Management, Harper & Row, 1965

14. Stohr, E.A. and Tanniru, M. R. "A Data Base for Operation Research Models," Policy Analysis and Information Systems V4 N2, Dec. 1980.

15. Thompson, J. Organizations in Action, Mc Graw Hill Book Company, 1967.

436:ECG:716:maf
79u455-551

ODA DISTRIBUTION LIST

Director, Engineering Psychology
Programs (Code 455)
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217 (5 cys)

Defense ~~Documentation~~ Technical Information Center
Cameron Station, Bldg. 5
Alexandria, VA 22314 (12 cys)

CDR Paul Chatelier
Office of the Deputy Under Secretary
of Defense
OUSDRE (E&LS)
Pentagon, Room 3D129
Washington, D.C. 20301

CAPT John Duncan
Office of the Secretary of Defense
(C3I)
Pentagon, Room 3C200
Washington, DC 20301

Dr. Craig Fields
Director, Cybernetics Technology
Office
Defense Advanced Research Projects
Agency
1400 Wilson Boulevard
Arlington, VA 22209

Office of the Chief of Naval
Operations, OP987H
Personnel Logistics Plans
Washington, DC 20350

Dr. A. L. Slafkosky
Scientific Advisor
Commandant of the Marine Corps
Code RD-1
Washington, DC 20380

Commanding Officer
ONR Eastern/Central Regional Office
ATTN: Dr. J. Lester
Bldg. 114, Section D
666 Summer Street
Boston, MA 02210

Commanding Officer
ONR Branch Office
ATTN: Dr. Charles Davis
536 South Clark Street
Chicago, IL 60605

Commanding Officer
ONR Western Regional Office
ATTN: Dr. E. Gloye
1030 East Green Street
Pasadena, CA 91106

Commanding Officer
ONR Western Regional Office
ATTN: Mr. R. Lawson
1030 East Green Street
Pasadena, CA 91106

Tactical Development & Evaluation Support
Code 230
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217

Naval Analysis Programs
Code 431
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217 (2 cys)

Operations Research Program
Code 434
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217

Statistics and Probability Program
Code 436
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217

Information Systems Program
Code 437
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217

436:ECG:716:maf
79u455-551

ODA DISTRIBUTION LIST

Dr. James McGrath
Code 311
Navy Personnel Research and
Development Center
San Diego, CA 92152

Management Support Department
Code 210
Navy Personnel Research and
Development Center
San Diego, CA 92152

Naval Electronics Systems Command
Human Factors Engineering Branch
Code 4701
Washington, DC 20360

Director
Naval Research Laboratory
Technical Information Division
Code 2627
Washington, DC 20375 (6 cys)

Mr. Arnold Rubinstein
Naval Material Command
NAVMAT 08D22
Washington, DC 20360

Commander, Naval Electronics
Systems Command
Command and Control Division
Code 530
Washington, DC 20360

Dr. John Silva
Head, Human Factors Division
Naval Ocean Systems Center
San Diego, CA 92152

Dr. Jesse Orlansky
Institute for Defense Analyses
400 Army-Navy Drive
Arlington, VA 22202

Human Factors Department
Code N215
Naval Training Equipment Center
Orlando, FL 32813

Dr. Gary Poock
Operations Research Department
Naval Postgraduate School
Monterey, CA 92940

Dr. Joseph Zeidner
Technical Director
U.S. Army Research Institute
5001 Eisenhower Avenue
Alexandria, VA 22333

Dr. Donald A. Topmiller
Chief, Systems Effectiveness
Branch
Human Engineering Division
Wright Patterson AFB, OH 45433

Dr. H. W. Sinaiko
Smithsonian Institution
801 N. Pitt Street
Alexandria, VA 22314

Office of the Chief of Naval
Operations OP942
Pentagon
Washington, DC 20350

Office of the Chief of Naval
Operations OP987C
R&D Plans Division
Washington, DC 20350

436:ECG:716:maf
79u455-551

ODA DISTRIBUTION LIST

Commander
Naval Electronics Systems Command
C3 Project Office
PME 108-1
Washington, DC 20360

CDR P. M. Curran
Human Factors Engineering Division
Code 604
Naval Air Development Center
Warminster, PA 18974

M. L. Metersky
Naval Air Development Center
Code 5424
Warminster, PA 18974

Dr. Edgar M. Johnson, Director
Organizations & Systems Research
Laboratory
U.S. Army Research Laboratory
5001 Eisenhower Avenue
Alexandria, VA 22333

Dr. David Dianich
Chairman, Dept. of Business and
Economics
Salisbury State College
Salisbury, MD 21801

Mr. Victor Monteleon
Naval Ocean Systems Center
San Diego, CA 92152

Commander, Naval Electronics
Systems Command
ELEX-03
Washington, DC 20360

CDR Richard Schlaff
NIPSSA
Hoffman Bldg. #1
2461 Eisenhower Avenue
Alexandria, VA 22331

Dr. Chantee Lewis
Management Department
Naval War College
Newport, RI 02840

Dr. John Shore
Naval Research Laboratory
Code 5403
Communications Sciences Division
Washington, DC 20375

Dr. Meredith Crawford
American Psychological Association
Office of Educational Affairs
1200 17th Street, NW.
Washington, DC 20036

Dr. William Dejka
ACCAT
Naval Ocean Systems Center
San Diego, CA 92152

Mr. Merlin Malehorn
Office of the Chief of Naval
Operations (Op 102)
Washington, DC 20350

Dr. S. D. Epstein
Analytics
2500 Maryland Road
Willow Grove, PA 19090

Dr. Amos Freedy
Perceptronics, Inc.
6271 Variel Avenue
Woodland Hills, CA 91364

Dr. G. Hurst
University of Pennsylvania
Wharton School
Philadelphia, PA 19174

436:ECG:716:maf
79u455-551

ODA DISTRIBUTION LIST

Dr. Miley Merkhofer
Stanford Research Institute
Decision Analysis Group
Menlo Park, CA 94025

Mr. George Pugh
Decision Science Applications, Inc.
1500 Wilson Boulevard
Arlington, VA 22209

Dr. Arthur Siegel
Applied Psychological Services
Science Center
404 E. Lancaster Street
Wayne, PA 19087

Mr. David Walsh
Integrated Sciences Corporation
1640 Fifth Street
Santa Monica, CA 90401

LCDR J. A. Sears
Department of MIS
College of Business Administration
University of Arizona
Tucson, AZ 85721

I. R. Mirman
Asst for Special Projects
HQ AFSC-DL
Andrews AFB, MD 20334

Mr. Joseph Wohl
MITRE Corporation
Box 208
Bedford, MA 01730

Dr. Kenneth Gardner
Applied Psychology Unit
Admiralty Marine Technology
Establishment
Teddington, Middlesex TW11 OLN
ENGLAND

Mr. Tim Gilbert
MITRE Corporation
1820 Dolly Madison Blvd
McLean, VA 22102

Mr. Leslie Innes
Defence & Civil Institute of
Environmental Medicine
P. O. Box 2000
Downsview, Ontario M3M 3B9
Canada

Dr. Rex Brown
Decision Science Consortium
Suite 421
7700 Leesburg Pike
Falls Church, VA 22043

Dr. A. C. Miller, III
Applied Decision Analysis
3000 Sand Hill Road
Menlo Park, CA 94025